# Temperature-Aware Task Allocation

The current trend of multi-core processors aims to dramatically improve the performance by exploiting parallelisms in the applications running on them. This performance improvement comes at the cost of thermal effects induced by high power densities such as temperature imbalances and hotspots, affecting the stability and the performance of the processor. Thermal-Aware task scheduling aims at optimizing the performance of multi-core processors. Use of Dynamic Thermal Management (DTM) through digital thermal sensors and dynamic voltage and frequency scaling (DVFS) is commonplace in many current generation processors.

We have used the event driven thermal estimator proposed in [June\_TCAD], to simulate the thermal – aware scheduling. The Event driven thermal estimator is based on power events. A power event is associated with increase or decrease in power generated by a core. The allocation or removal of a task from a core generates a power event with instantaneous power change at the beginning and end of the task. We determine the power events by profiling the power consumption of every task. For simulation purposes, we carry out a static (offline) power profiling.

The estimator models each of the cores in the multi-core processor as a thermal RC network, described by a set of linear ordinary differential equations. This thermal model is similar to the one used in HotSpot simulator and makes it a good choice for high level thermal estimation. For a many-core system, the relation between power, temperatures, thermal resistance and capacitance is expressed as -



HotSpot solves the thermal model at every regular interval making it computationally extensive for dynamic thermal management. We make use of a look-up table approach as described in [June\_TCAD]. A LUT is used to model the thermal characteristics of the processor chip. The LUT is modeled to be a three dimensional matrix consisting of multiple tables. Each table has the temperature trace of the chip taken at regular intervals when one watt of power is injected in one particular core leaving the other cores untouched. Each table in the LUT corresponds to power injection in one particular core. The LUT models the approximate thermal characteristics of each core for 1 Watt power injection.

The task allocation for each bench mark, the power injected into each core subsequently and the communication between the cores is obtained from the Gem5 Simulator.

The task allocations and the powers for each core are modeled as power events to estimate a thermal map based on the LUT. We then run a task reallocation heuristic based on the future temperature trends.

These future temperatures can be estimated using the LUT. The future thermal map at a future time *tf* can be calculated based on the current thermal map  at time *tc* by adding the temperature increment of each core in the interval *Δt* to the current thermal map as follows -



where ,  and are N-element vectors denoting the future thermal mape, the current thermal map and the temperature increment map of each of the N cores respectively.

The temperature increment is calculated as follows –



where, ae is an atomic power event in the list of events E, LUTrow denotes a row of the LUT for time tf and tc respectively for the LUT of core given by ae.core, the core where the power event occurs.

We then run the Future Temperature Trends (FTT) heuristic to obtain a thermal aware task allocation. The FTT heuristic algorithm classifies all the idle cores into two sets – Core+ for temperature-increasing and Core- for temperature-decreasing, based on the difference in current and predicted temperature for that core. The weights are assigned as follows –



where, a+ is the temperature increment and a- is the temperature decrement. We then choose from each set the core with the minimum weight and randomly allocate the task to one of the cores selected.

We also run another heuristic which modifies the above one to incorporate the network characteristics. For each task to be allocated, we calculate the weights using the Future Temperature Trend. After calculating the weights, we create the selection set in the ascending order of weights of the cores. For the network characteristics, we take into consideration the cores which have been allocated a particular task only. Let Alloc\_Set be a set of all cores which have been allocated a task already. For each core in the selection set, another weight is calculated as follows –



where, *j* is a core from Alloc\_Set, *hij* is the hop-count from core *i* to core *j* and *fij* is the frequency of communication between core *i* and *j*. The task is allocated to the core with the smallest weight so obtained.

After all the tasks are reallocated, the results are input to HotSpot simulator to obtain a temperature profile for the allocation.